MULTI-INPUT MULTI-OUTPUT FAST QR DECOMPOSITION ALGORITHM FOR ACTIVE NOISE CONTROL

Mobien Shoaib,[1,3] *Stefan Werner,*[1] *José A. Apolinário Jr.,*[2] *and Saleh A. Alshebieli*[3]

[1]**Helsinki University of Technology**
Dept. of Sig. Proc. and Acoustics
Espoo, Finland
email: {mobien.shoaib, stefan.werner}@tkk.fi

[2]**Military Institute of Engineering**
Department of Electrical Engineering
Rio de Janeiro, Brazil
email: apolin@ieee.org

[3]**Prince Sultan Advanced Technology Research Inst. (PSATRI)**
King Saud University
Riyadh, Saudi Arabia
email: dsaleh@ksu.edu.sa

ABSTRACT

Fast QR-decomposition based recursive least-squares (FQRD-RLS) algorithms are known for their good numerical properties and low computational complexity. However, they have so far not been used in active noise control (ANC) because the general implementation would require multiple-input multiple-output (MIMO) FQRD-RLS algorithms, which are currently not available in the technical literature. Another reason is the lack of an explicit weight vector update equation in the FQRD-RLS algorithms, which prevents their use in structures where a copy of the coefficients is filtering a different input sequence than that of the adaptive filter, e.g., the modified filtered-x ANC (MFX-ANC) structure. In this paper, we derive a MIMO-FQRD-RLS algorithm based on backward prediction error updates. The proposed algorithm is applied to a multichannel MFX-ANC structure. We show how to avoid the explicit use of the weight vector in the MFX-ANC structure by reproducing the filtered-x signal from the internal variables of the proposed algorithm. Simulation results confirm that the learning curves of the MIMO-FQRD-RLS algorithm are identical to those obtained by the QRD-RLS algorithm.

1. INTRODUCTION

The family of QR-decomposition (QRD) based least-squares (LS) adaptive filters are known for fast convergence and stability in finite precision. Several low complexity versions of the QRD-LS algorithm have been derived in past, e.g., the QRD least-squares lattice and the fast QRD recursive least-squares (FQRD-RLS) algorithms [1]. The main disadvantage of these low-complexity versions is that they lack an explicit weight vector term, limiting themselves to problems seeking an estimate of the output error signal. It was recently shown in [2, 3] how to identify (or extract) the transversal weights embedded in the FQRD-RLS variables, thus enabling their use for system identification. Furthermore, in [4], it was shown how to use the FQRD-RLS variables for fixed filtering, e.g., burst-trained equalizers, without explicit knowledge of the associated weight vector.

Despite the recent progress, there are still applications where the available FQRD-RLS algorithms cannot be directly employed. To the best of our knowledge, it does not exist in literature an FQRD-RLS algorithm that takes multiple inputs and generates multiple outputs, i.e., a MIMO-FQRD-RLS algorithm. In this paper we develop a MIMO version of the numerically robust FQRD-RLS algorithm based on the update of backward prediction errors [5]. The new algorithm is applied to the problem of multichannel active noise control (ANC) using the modified filtered-x (MFX) structure [6].

In the MFX-ACN structure, a copy of the adaptive filter weights is used for filtering a different input sequence than used for adaptation. The resulting output sequence is referred to as the filtered-x signal. By extending the ideas in [4] we may reproduce the filtered-x signal exactly without the explicit knowledge of the MIMO-FQRD-RLS filter weights. This should be compared with the lattice based FQRD algorithm used for ANC in [6], which solved this problem by explicitly identifying the required coefficient vector. To reduce the complexity of such solution, it was suggested in [6] to identify and copy the weights on a periodical basis (i.e., not every iteration). Our approach avoids such an approximate solution while still maintaining a low computational complexity.

Simulation results are provided in order to validate the proposed method. A QR-decomposition based RLS algorithm is used as a benchmark. The total number of multiplications for each algorithm are listed in a table.

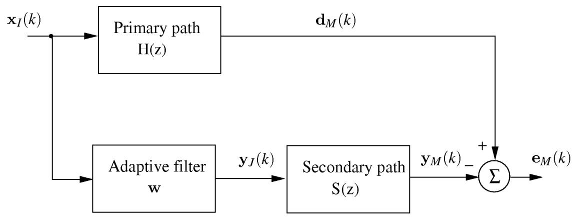

Figure 1: A simplified illustration of an ANC system.

2. ACTIVE NOISE CONTROL SYSTEM BASICS

This paper considers the MFX structure detailed in [6]. In the following we describe the basic multichannel ANC setup and its associated least-squares solution. Thereafter, we provide the necessary QRD-RLS equations and illustrate how the MFX-ANC structure becomes the natural implementation for QRD-RLS algorithms.

2.1 Simplified ANC setup and least-squares solution

The concept of a simplified multichannel ANC is shown in Fig. 1. An ANC cancels the unwanted noise by generating an anti-noise signal. The noise signal to be canceled $\mathbf{x}_I(k) \in \mathbb{R}^{I\times 1}$ is obtained by the ANC system using I "reference microphones". The ANC system processes the reference signal to generate the anti-signal of the reference $\mathbf{y}_J(k) \in \mathbb{R}^{J\times 1}$ using J actuators, where the jth element (output) is given as

$$y_j(k) = \sum_{i=1}^{I} \mathbf{x}_i^{\mathrm{T}}(k)\mathbf{w}_{i,j}, \tag{1}$$

where $\mathbf{x}_i(k) = [x_i(k) \cdots x_i(k-L+1)]^{\mathrm{T}} \in \mathbb{R}^{L\times 1}$ is the input vector of the ith channel, and $\mathbf{w}_{i,j} \in \mathbb{R}^{L\times 1}$ is the coefficient vector that links the ith input to jth output.

The M "error microphones" pick the de-noised signal and serve as a feedback to the ANC system. The path between the reference microphone and the error microphone is called the $I \times M$-MIMO primary path $\mathbf{H}(z)$, while the path between the speakers and the error microphone is called the $J \times M$-MIMO secondary path $\mathbf{S}(z)$. The output of the primary path is $\mathbf{d}_M(k)$, where the mth output is given as

$$d_m(k) = \sum_{i=1}^{I} \bar{\mathbf{x}}_i^{\mathrm{T}}(k)\mathbf{h}_{i,m} + n_M(k) \quad (2)$$

where where $\bar{\mathbf{x}}_i(k) = [x_i(k) \cdots x_i(k-L_p+1)]^{\mathrm{T}} \in \mathbb{R}^{L_p \times 1}$ is the input vector of the ith channel, , and $\mathbf{h}_{i,m} \in \mathbb{R}^{L_p \times 1}$ are the coefficients of the primary path between the ith input and mth output, assumed stationary for this discussion, and $n_m(k)$ is the additive noise. Similarly, the output of the secondary path is $\mathbf{y}_M(k) \in \mathbb{R}^{M \times 1}$, where the mth output is

$$\bar{y}_m(k) = \sum_{j=1}^{J} \mathbf{y}_j^{\mathrm{T}}(k)\mathbf{s}_{j,m}, \quad (3)$$

where $\mathbf{y}_j = [y_j(k) \cdots y_j(k-L_s+1)] \in \mathbb{R}^{L_s \times 1}$, and $\mathbf{s}_{i,m}(k) \in \mathbb{R}^{L_s \times 1}$ are the coefficients of the secondary path between the jth input from the adaptive filter and mth output. Finally, the error signal at the M error sensors is given by,

$$\mathbf{e}_M(k) = \mathbf{d}_M(k) - \mathbf{y}_M(k) \quad (4)$$

where $\mathbf{y}_M(k) = [\bar{y}_1(k) \cdots \bar{y}_M(k)]^{\mathrm{T}} \in \mathbb{R}^{M \times 1}$. We see from (4) that signal $\mathbf{e}_M(k)$ measured by the error microphone array in Fig. 1 is the result of an acoustic subtraction of the desired signal vector $\mathbf{d}_M(k)$ and the output of the secondary path $\mathbf{y}_M(k)$. Let the adaptive filter, after convergence, and the secondary path in Fig. 1 be linear time-invariant systems. Then the secondary path can be placed before the adaptive filtering block without any loss of generality. As a result, the multichannel input signal $\mathbf{x}_I(k)$ is filtered with the secondary path before being used for the adaptive filtering. This can be carried out by defining $x_{m,r}(k), 1 \leq i \leq R$ as one of the $R = IJ$ possible values of

$$x_{m,\{i,j\}}(k) = \tilde{\mathbf{x}}_i^{\mathrm{T}}(k)\hat{\mathbf{s}}_{j,m} \quad (5)$$

where $x_{m,\{i,j\}}(k)$ is the filtered input signal, $\hat{\mathbf{s}}_{j,m} \in \mathbb{R}^{L_s \times 1}$ is the estimate of the secondary path for jth input and mth output, and $\tilde{\mathbf{x}}_i(k) \in \mathbb{R}^{L_s \times 1}$ is the ith channel input from the reference microphones. The definition in (3) has to be changed, therefore the mth output of the adaptive filter using the filtered input signal is given by $\bar{y}_m(k)$and is now written as

$$\bar{y}_m(k) = \mathbf{x}_m^{\mathrm{T}}(k)\mathbf{w} \quad (6)$$

where

$$\mathbf{x}_m(k) = [\mathbf{x}_{m,R}^{\mathrm{T}}(k) \cdots \mathbf{x}_{m,R}^{\mathrm{T}}(k-L+1)]^{\mathrm{T}} \in \mathbb{R}^{RL \times 1} \quad (7)$$

$$\mathbf{x}_{m,R}(k) = [x_{m,1}(k)\, x_{m,2}(k) \cdots x_{m,R}(k)]^{\mathrm{T}} \in \mathbb{R}^{R \times 1} \quad (8)$$

and vector $\mathbf{w} \in \mathbb{R}^{RL \times 1}$ is formed by stacking the $R = IJ$ vectors $\mathbf{w}_{i,j}$ defined in (1). By employing the definition of the output signal (6) in the error expression (4), $\mathbf{w}$ is chosen to minimize the following deterministic weighted least-squares cost function

$$\xi_D(k) = \sum_{i=0}^{k} \lambda^{k-i} ||\mathbf{d}_M(i) - \mathbf{X}_M^{\mathrm{T}}(i)\mathbf{w}||^2 \quad (9)$$

where λ is the forgetting factor and

$$\mathbf{X}_M(k) = \left[\mathbf{X}_{M,R}^{\mathrm{T}}(k) \cdots \mathbf{X}_{M,R}^{\mathrm{T}}(k-L+1)\right]^{\mathrm{T}} \in \mathbb{R}^{RL \times M} \quad (10)$$

$$\mathbf{X}_{M,R}(k) = [\mathbf{x}_{1,R}(k)\, \mathbf{x}_{2,R}(k) \cdots \mathbf{x}_{M,R}(k)]^{\mathrm{T}} \in \mathbb{R}^{M \times R}. \quad (11)$$

The optimal solution at time instant k, obtained by differentiating (9) with respect to $\mathbf{w}$, is given by $\mathbf{w}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k)$, where

$$\mathbf{R}(k) = \sum_{i=0}^{k} \lambda^{k-i} \mathbf{X}_M(i)\mathbf{X}_M^{\mathrm{T}}(i) \in \mathbb{R}^{RL \times RL}$$
$$\mathbf{p}(k) = \sum_{i=0}^{k} \lambda^{k-i} \mathbf{X}_M(i)\mathbf{d}_M(i) \in \mathbb{R}^{RL \times 1}. \quad (12)$$

The recursive least-squares (RLS) coefficient update for this filtered-x structure is

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \mathbf{R}^{-1}(k)\mathbf{X}_M(k)\mathbf{e}_M(k) \quad (13)$$

The inverse $\mathbf{R}^{-1}(k)$ can be obtained recursively in terms of $\mathbf{R}^{-1}(k-1)$ using the *matrix inversion lemma*. We see that the filtered-x (FX) RLS algorithm in (13) requires only knowledge of the input-signal matrix $\mathbf{X}_M(k)$ and the measured error vector $\mathbf{e}_M(k)$, whereas the unknown desired vector $\mathbf{d}_M(k)$ is not needed.

2.2 Modified filtered-x ANC setup and QRD-RLS equations

The main problem with the FX-RLS algorithm is potential divergence behavior in finite precision and high computational complexity. The QRD-RLS and, its fast version, the FQRD-RLS algorithm could provide a more robust solution with reduced complexity. However, both these algorithms require explicit knowledge of $\mathbf{d}_M(k)$ to minimize the objective function in (9). This should be compared with the FX-RLS algorithm of previous section that directly employs vector $\mathbf{e}_M(k)$ measured by the error microphones. However, we see from Fig. 1 that if we pass the actuator signal $\mathbf{y}_J(k)$ through an off-line estimate of the secondary-path channel $\hat{\mathbf{S}}(z)$, an estimate $\hat{\mathbf{d}}(k)$ of the desired signal vector can be obtained, whose elements are given by

$$\hat{d}_m(k) = e_m(k) + \hat{y}_m(k) \quad (14)$$

where $\hat{y}_m(k) = \sum_{j=1}^{J} \mathbf{y}_j^{\mathrm{T}}(k)\hat{\mathbf{s}}_{j,m}(k)$ and $\hat{\mathbf{s}}_{j,m}(k)$ being the estimate of $\mathbf{s}_{j,m}(k)$ in $\mathbf{S}(z)$. Taking this QRD approach leads to the realization depicted in Fig. 2, which is known as the MFX structure. Equipped with an estimate of $\mathbf{d}_M(k)$ through (14), we may derive MIMO-QRD-RLS algorithms for ANC. In the following we provide the basic QRD-RLS equations used for deriving the MIMO-FQRD-RLS version in Section 3. We also show in Section 3 how to obtain the output $\mathbf{y}_J(k)$ in Fig. 2 without explicit knowledge of the MIMO-FQRD-RLS weights $\mathbf{w}(k)$.

Considering a QRD-RLS algorithm for obtaining vector $\mathbf{w}$, corresponding to an estimate of the vectorized $I \times J$-MIMO channel $\mathbf{H}(z)$, (9) is re-written as the norm of an increasing order vector

$$\mathbf{e}(k) = \begin{bmatrix} \mathbf{d}_M(k) \\ \lambda^{1/2}\mathbf{d}_M(k-1) \\ \vdots \\ \lambda^{k/2}\mathbf{d}_M(0) \end{bmatrix} - \begin{bmatrix} \mathbf{X}_M^T(k)\mathbf{w} \\ \lambda^{1/2}\mathbf{X}_M^T(k-1)\mathbf{w} \\ \vdots \\ \lambda^{k/2}\mathbf{X}_M^T(0)\mathbf{w} \end{bmatrix} = \mathbf{d}(k) - \mathbf{X}(k)\mathbf{w} \quad (15)$$

where $\mathbf{e}(k) \in \mathbb{R}^{M(k+1) \times 1}$ is the error vector, $\mathbf{d}(k) \in \mathbb{R}^{M(k+1) \times 1}$ is the desired signal vector, and $\mathbf{X}(k) \in \mathbb{R}^{M(k+1) \times RL}$ is the input data matrix encompassing values for all time instances from 0 up to k.

In the QRD-RLS algorithm, the triangularization of the input data matrix is obtained with

$$\begin{bmatrix} \mathbf{0}_{(M(k+1)-RL) \times RL} \\ \mathbf{U}(k) \end{bmatrix} = \mathbf{Q}(k)\mathbf{X}(k) \quad (16)$$

where $\mathbf{Q}(k) \in \mathbb{R}^{M(k+1) \times M(k+1)}$ is the Givens rotation matrix, and $\mathbf{U}(k) \in \mathbb{R}^{RL \times RL}$ is a triangular matrix known as the Cholesky factor of the input data matrix. Applying the Givens rotation matrix to $\mathbf{e}(k)$ results in

$$\mathbf{Q}(k)\mathbf{e}(k) = \begin{bmatrix} \mathbf{d}_{q1}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} - \begin{bmatrix} \mathbf{0}_{(M(k+1)-RL) \times RL} \\ \mathbf{U}(k) \end{bmatrix} \mathbf{w} \quad (17)$$

From (17), we are able to minimize the norm of $\mathbf{e}(k)$, at time instant k, by choosing $\mathbf{w}$ as

$$\mathbf{w} = \mathbf{U}^{-1}(k)\mathbf{d}_{q2}(k) \quad (18)$$

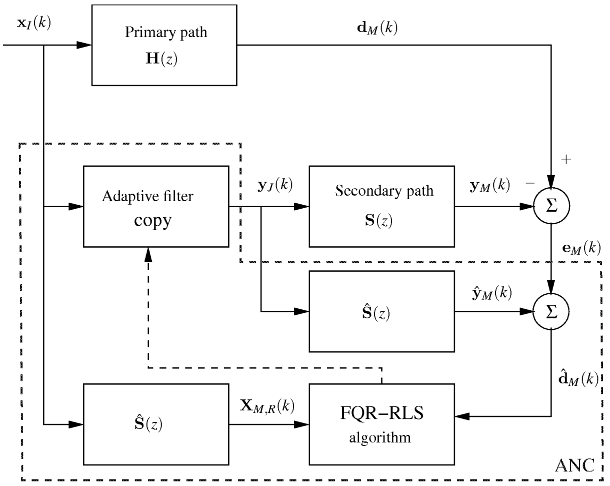

Figure 2: FQRD-RLS in an active noise control (ANC) system.

In QRD-RLS algorithm the update for the Cholesky factor matrix and the rotated desired signal vector are provided as follows

$$\begin{bmatrix}\mathbf{0}_{M\times RL}\\ \mathbf{U}(k)\end{bmatrix} = \mathbf{Q}_\theta \begin{bmatrix}\mathbf{X}_M^{\mathrm{T}}(k)\\ \lambda^{1/2}\mathbf{U}(k-1)\end{bmatrix} \quad (19)$$

$$\begin{bmatrix}\mathbf{e}_{qM}(k)\\ \mathbf{d}_{q2}(k)\end{bmatrix} = \mathbf{Q}_\theta \begin{bmatrix}\mathbf{d}_M(k)\\ \lambda^{1/2}\mathbf{d}_{q2}(k-1)\end{bmatrix} \quad (20)$$

where $\mathbf{Q}_\theta(k) \in \mathbf{R}^{(M+RL)\times(M+RL)}$ is the Givens rotation matrix that annihilates the values of $\mathbf{X}_M$ when used to update $\mathbf{U}(k)$ and is defined by the partition matrix

$$\mathbf{Q}_\theta(k) = \begin{bmatrix}\Gamma(k) & \mathbf{G}(k)\\ \mathbf{F}(k) & \mathbf{E}(k)\end{bmatrix} \quad (21)$$

where

$$\begin{aligned}
\Gamma(k) &= \begin{bmatrix}\gamma_1(k) & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \gamma_M(k)\end{bmatrix}\\
\mathbf{G}(k) &= -\Gamma(k)\mathbf{A}(k) = -\Gamma(k)\lambda^{-1/2}\mathbf{U}^{-T}(k-1)\mathbf{X}_M(k)\\
\mathbf{F}(k) &= \mathbf{U}^{-T}(k)\mathbf{X}_M(k)\\
\mathbf{E}(k) &= \lambda^{1/2}\mathbf{U}^{-T}(k)\mathbf{U}^{T}(k-1)
\end{aligned} \quad (22)$$

Finally, the *a posteriori* error vector, $\mathbf{e}_M(k)$ computed with $\mathbf{w}(k)$, is given as

$$\mathbf{e}_M(k) = \Gamma(k)\mathbf{e}_{qM}(k). \quad (23)$$

3. MIMO-FQRD-RLS ALGORITHM FOR MFX ANC

In this section we derive a MIMO-FQRD-RLS algorithm that renders a computationally efficient and robust implementation of the MFX-ANC structure. Common to the FQRD-RLS algorithms already available in the literature, the proposed algorithm does not have explicit update equations for vector $\mathbf{w}(k)$. However, as earlier mentioned, the weights embedded in the MIMO-FQRD-RLS algorithm are needed to reproduce output signal $\mathbf{y}_J(k)$ in Fig. 2. We know from [2] that it should be possible to acquire the weights through a sequential weight extraction procedure. However, such an approach would lead to a solution of $\mathscr{O}(R^2L^2)$ complexity per iteration. In other words, there would not be any obvious advantage of using an MIMO-FQRD-RLS solution in place of, say, an Inverse QRD-RLS solution [5]. To solve this problem, we also show in this section how to reproduce the exact output $\mathbf{y}_J(k)$ by conveniently reusing variables from the proposed MIMO-QRD-RLS algorithm.

3.1 Derivation of the MIMO-FQRD-RLS algorithm

The fast versions of the QRD-RLS algorithm based on update of the backward prediction errors are known for their numerical stability and its *a posteriori* version (FQRD-RLS_POS_B) is considered here for ANC application. This algorithm offers low complexity solution by using matrix of smaller dimensions in the updating procedure of inner variables, unlike the conventional QRD-RLS or the Inverse QRD-RLS algorithms. For the FQRD-RLS PRI B algorithm, $\mathbf{F}(k) \in \mathbb{R}^{RL\times M}$ in (22) is considered for the update. As a result, the problem of updating the $RL\times RL$ matrix $\mathbf{U}(k)$ is reduced to updating a $RL\times M$ matrix, where $M << RL$.

We start the derivation of the fast version by considering the extended input data matrix defined as

$$\mathbf{X}^{L+1}(k) = \begin{bmatrix}\mathbf{D}_f(k) & \begin{matrix}\mathbf{X}(k-1)\\ \mathbf{0}_{M\times RL}\end{matrix}\end{bmatrix} = [\mathbf{X}(k) \quad \mathbf{D}_b(k)] \quad (24)$$

where $\mathbf{D}_f(k) \in \mathbb{R}^{M(k+1)\times R}$ is the desired signal vector in a forward prediction

$$\mathbf{D}_f(k) = \begin{bmatrix}\mathbf{X}_{M,R}(k)\\ \vdots\\ \lambda^{k/2}\mathbf{X}_{M,R}(0)\end{bmatrix}, \quad (25)$$

and $\mathbf{d}_b(k) \in \mathbb{R}^{M(k+1)\times R}$ is the desired signal vector in a backward prediction

$$\mathbf{D}_b(k) = \begin{bmatrix}\mathbf{X}_{M,R}(k-RL)\\ \vdots\\ \lambda^{(k-RL)/2}\mathbf{X}_{M,R}(0)\\ \mathbf{0}_{ML\times R}\end{bmatrix}. \quad (26)$$

In order to triangularize (24) and obtain $\mathbf{U}^{L+1}(k)$, three sets of Givens rotation matrices are needed [7]. Note that we have introduced a few rows of zeros below $\mathbf{X}^{L+1}(k)$ in order to allow the formation of a triangular extended Cholesky factor of dimension $(RL+R)\times(RL+R)$.

$$\begin{aligned}
\mathbf{Q}'_f(k)\mathbf{Q}_f(k)\begin{bmatrix}\mathbf{Q}(k-1) & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{R\times R}\end{bmatrix}\begin{bmatrix}\mathbf{X}^{L+1}(k)\\ \mathbf{0}_{(R-M)\times(RL+R)}\end{bmatrix} &=\\
= \quad \mathbf{Q}'_f(k)\mathbf{Q}_f(k)\begin{bmatrix}\mathbf{E}_{fq1}(k) & \mathbf{0}\\ \mathbf{D}_{fq2}(k) & \mathbf{U}(k-1)\\ \mathbf{X}_{M,R}(0) & \mathbf{0}_{M\times RL}\\ \multicolumn{2}{c}{\mathbf{0}_{(R-M)\times(RL+R)}}\end{bmatrix}&\\
= \quad \mathbf{Q}'_f(k)\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{D}_{fq2}(k) & \mathbf{U}(k-1)\\ \mathbf{E}_f(k) & \mathbf{0}\end{bmatrix}&
\end{aligned} \quad (27)$$

From the previous expression and using the fixed-order matrices $\mathbf{Q}_\theta(k-1)$ embedded in $\mathbf{Q}(k-1)$ and $\overline{\mathbf{Q}}_f(k)$ embedded in $\mathbf{Q}_f(k)$, it is possible to obtain the following equations.

$$\begin{bmatrix}\mathbf{E}_{fqM}^{\mathrm{T}}(k)\\ \mathbf{D}_{fq2}(k)\end{bmatrix} = \mathbf{Q}_\theta(k-1)\begin{bmatrix}\mathbf{X}_{M,R}(k)\\ \lambda^{1/2}\mathbf{D}_{fq2}(k)\end{bmatrix} \quad (28)$$

$$\begin{bmatrix}\mathbf{0}^T\\ \mathbf{E}_f(k)\end{bmatrix} = \overline{\mathbf{Q}}_f(k)\begin{bmatrix}\mathbf{E}_{fqM}^T(k)\\ \lambda^{1/2}\mathbf{E}_f(k-1)\end{bmatrix} \quad (29)$$

In (28), $\mathbf{E}_{fqM}(k)$ corresponds to the first M rows of $\mathbf{E}_{fq}(k)$, the *rotated forward error*, and, in (29), $\mathbf{E}_f(k)$ is the $R\times R$ *forward prediction error covariance* (lower-triangular) matrix.

Removing the ever-increasing null section in (27) and using the fixed-order matrix $\mathbf{Q}'_{\theta f}(k)$ embedded in $\mathbf{Q}'_f(k)$, we obtain

$$\bar{\mathbf{U}}^{L+1}(k) = \mathbf{Q}'_{\theta f}(k)\begin{bmatrix}\mathbf{D}_{fq2}(k) & \mathbf{U}(k-1)\\ \mathbf{E}_f(k) & \mathbf{0}\end{bmatrix}. \quad (30)$$

A natural choice for $\mathbf{Q}'_{\theta f}(k)$ could be the Givens rotation matrix that annihilate $\mathbf{D}_{fq2}(k)$ against $\mathbf{E}_f(k)$ while keeping its triangular structure. Also from (30), we can write the following equation where $\mathbf{E}_f^0(k)$ was named the *zero-order error covariance matrix* in [7].

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{E}_f^0(k) \end{bmatrix} = \mathbf{Q}'_{\theta f}(k) \begin{bmatrix} \mathbf{D}_{fq2}(k) \\ \mathbf{E}_f(k) \end{bmatrix} \tag{31}$$

Since we are interested in deriving the *a posteriori* version of the FQRD-RLS algorithm, we update matrix $\mathbf{F}^{L+1}(k)$ as follows, i.e., based on the inverse of the extended order Cholesky factor obtained in (30). Note that, similarly to the case of employing forward prediction reference signal in the definition of the extended order input matrix, using backward prediction reference signal leads to the conclusion that the last RL rows of $\mathbf{F}^{L+1}(k)$ correspond to $\mathbf{F}(k)$. By replacing the definition of $\mathbf{X}_M(k)$ in (10) and the inverse transposed of Cholesky factor given by (30), we have:

$$\mathbf{F}^{L+1}(k) = \begin{bmatrix} * \\ \mathbf{F}(k) \end{bmatrix} = \mathbf{Q}'_{\theta f}(k) \begin{bmatrix} \mathbf{F}(k-1) \\ \mathbf{F}_p(k) \end{bmatrix}, \tag{32}$$

where $\mathbf{F}_p(k)$ correspond to the product of $\mathbf{E}_f^{-T}(k)$ with the forward prediction error matrix $\mathbf{E}_{fM}(k)$. The computation of $\mathbf{F}_p(k)$, however, is quite simplified with the help of the following expression (the proof is omitted here due to lack of space but similar derivation can be found in [7]).

$$\overline{\mathbf{Q}}_f(k) \begin{bmatrix} \mathbf{\Gamma}(k) \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} * \\ \mathbf{F}_p(k) \end{bmatrix} \tag{33}$$

where the asterisks in the two previous equations represent non-zero values that are not used in this algorithm.

Once knowing the structure of matrix $\mathbf{Q}_\theta(k)$, its rotation angles can be obtained using

$$\mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}(\mathbf{k}) \\ \mathbf{F}(k) \end{bmatrix}. \tag{34}$$

Although the *a posteriori* error vector was given in (23), the *a priori* errors $\varepsilon_M(k)$ can be obtained as follows. Note that matrix $\mathbf{\Gamma}(k)$ is diagonal and its inverse is easily obtained. The equations of the *a posteriori* version of the MIMO Fast QRD-RLS algorithm are summarized in Table 1, and the computational complexity is given in Table 3.

$$\varepsilon_M(k) = \mathbf{\Gamma}^{-1}(k)\mathbf{e}_{qM}(k) \tag{35}$$

3.2 Re-producing the filtered-x signal $\mathbf{y}_J(k)$

While running the MIMO-FQRD-RLS algorithm in the lower branch of Fig. 2, we need to have access to the associated coefficient vector for reproducing the output signal $\mathbf{y}_J(k)$. As noted earlier, making the coefficient vector explicitly available using the technique in [2] is not an attractive solution considering its computational complexity. One can avoid this complexity problem by extracting and copy the weights at a reduced rate, say once every K samples. This type of solution was considered in [6] for the QRD least-squares lattice (QRD-LSL) algorithm. Such approach certainly affects the convergence behavior of the adaptive implementation. Our goal is to reproduce, at each iteration, the exact output $\mathbf{y}_J(k)$ that corresponds to the weights embedded in the MIMO-FQRD-RLS algorithm.

Similar to definition of $\mathbf{y}_M(k)$ in (15) we may express $\mathbf{y}_J(k)$ as

$$\mathbf{y}_J(k) = \mathbf{X}_J^{\mathrm{T}}(k)\mathbf{w}(k) = \mathbf{X}_J^{\mathrm{T}}(k)\mathbf{U}^{-1}(k)\mathbf{d}_{q2}(k) \tag{36}$$

where

$$\mathbf{X}_J(k) = [\mathbf{x}_1(k) \cdots \mathbf{x}_J(k)] \in \mathbb{R}^{\{R=IJ\}L\times J}, \tag{37}$$

Table 1: MIMO-FQRD_POS_B Algorithm

For each k, do

1. Obtaining $\mathbf{D}_{fq2}(k)$ and $\mathbf{E}_{fqM}(k)$
$$\begin{bmatrix} \mathbf{E}_{fqM}^T(k) \\ \mathbf{D}_{fq2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k-1) \begin{bmatrix} \mathbf{X}_{M,R}^{\mathrm{T}}(k) \\ \lambda^{1/2}\mathbf{D}_{fq2}(k) \end{bmatrix}$$
2. Obtaining $\mathbf{E}_f(k)$ and $\overline{\mathbf{Q}}_f(k)$
$$\begin{bmatrix} \mathbf{0}^T \\ \mathbf{E}_f(k) \end{bmatrix} = \overline{\mathbf{Q}}_f(k) \begin{bmatrix} \mathbf{E}_{fqM}^T(k) \\ \lambda^{1/2}\mathbf{E}_f(k-1) \end{bmatrix}$$
3. Obtaining $\mathbf{F}_p(k)$
$$\begin{bmatrix} * \\ \mathbf{F}_p(k) \end{bmatrix} = \overline{\mathbf{Q}}_f(k) \begin{bmatrix} \mathbf{\Gamma}(k) \\ \mathbf{0} \end{bmatrix}$$
4. Obtaining $\mathbf{Q}'_{\theta f}(k)$
$$\begin{bmatrix} \mathbf{0} \\ \mathbf{E}_f^0(k) \end{bmatrix} = \mathbf{Q}'_{\theta f}(k) \begin{bmatrix} \mathbf{D}_{fq2}(k) \\ \mathbf{E}_f(k) \end{bmatrix}$$
5. Obtaining $\mathbf{F}(k)$, the last LR rows of $\mathbf{F}^{L+1}(k)$
$$\mathbf{F}^{L+1}(k) = \mathbf{Q}'_{\theta f}(k) \begin{bmatrix} \mathbf{F}(k-1) \\ \mathbf{F}_p(k) \end{bmatrix}$$
6. Obtaining $\mathbf{Q}_\theta(k)$ and $\mathbf{\Gamma}(k)$
$$\mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}(\mathbf{k}) \\ \mathbf{F}(k) \end{bmatrix}$$
7. Joint Estimation
$$\begin{bmatrix} \mathbf{e}_{qM}(k) \\ \mathbf{d}_{q2}(k) \end{bmatrix} = \mathbf{Q}_\theta(k) \begin{bmatrix} \mathbf{d}_M(k) \\ \lambda^{1/2}\mathbf{d}_{q2}(k-1) \end{bmatrix}$$
8. Obtaining the *a posteriori* error
$$\mathbf{e}_M(k) = \mathbf{\Gamma}(k)\mathbf{e}_{qM}(k)$$

}

and $\mathbf{x}_j(k) \in \mathbb{R}^{RL\times 1}$ is obtained by stacking the columns of the matrix $\mathbf{X}^j(k)$, which is defined as,

$$\mathbf{X}^j(k) = \begin{bmatrix} \mathbf{0}_{j-1\times IL} \\ \mathbf{x}_I^{\mathrm{T}}(k) \cdots \mathbf{x}_I^{\mathrm{T}}(k-L+1) \\ \mathbf{0}_{J-j\times IL} \end{bmatrix} \tag{38}$$

where $\mathbf{x}_I(k) = [x_1(k) \cdots x_I(k)] \in \mathbb{R}^{I\times 1}$ is the input data vector of I channels. Note that the output $\mathbf{y}_J(k)$ is defined differently than in (1), as in MIMO-FQRD-RLS algorithm the coefficient of all the channels are only available as $\mathbf{w}(k)$. Matrix $\mathbf{U}^{-1}(k)$ and vector $\mathbf{d}_{q2}(k)$ in (36) are parameters of the MIMO-FQRD-RLS algorithm running in the lower branch of Fig. 2. Vector $\mathbf{d}_{q2}(k)$ is directly available from the MIMO-FQRD-RLS algorithm. However, $\mathbf{U}^{-1}(k)$ is hidden in $\mathbf{F}(k) = \mathbf{U}^{-\mathrm{T}}(k)\mathbf{X}_M(k)$ while our objective is to construct and update

$$\mathbf{F}_J(k) = \mathbf{U}^{-\mathrm{T}}(k)\mathbf{X}_J(k) \in \mathbb{R}^{RL\times J}. \tag{39}$$

where we may note the use of a different input matrix. Since matrices $\mathbf{X}_M(k)$ and $\mathbf{X}_J(k)$ are both initially set to zero, we may exploit (32) in order to update $\mathbf{F}_J(k)$. The Givens rotation matrix and the matrix $\mathbf{E}_f(k)$ are copied from the MIMO-FQRD-RLS algorithm, while the forward prediction error matrix is computed using the current inputs. The procedure is summarized in Lemma 1.

Lemma 1. *Let $\mathbf{U}^{-T}(k) \in \mathbb{R}^{RL\times RL}$ denote the upper triangular inverse transposed Cholesky factor matrix in Table* 1. *Given $\mathbf{Q}_{\theta f}(k) \in \mathbb{R}^{R(L+1)\times R(L+1)}$, $\mathbf{D}_{fq2}(k) \in \mathbb{R}^{RL\times R}$, and $\mathbf{E}_f(k) \in \mathbb{R}^{R\times R}$ from the MIMO-FQRD-RLS algorithm operating in the lower branch of ANC, then we can obtain $\mathbf{F}_J(k)$ from $\mathbf{F}_J(k-1)$ using*

$$\begin{bmatrix} * \\ \mathbf{F}_J(k) \end{bmatrix} = \mathbf{Q}_{\theta f}(k) \begin{bmatrix} \mathbf{F}_J(k-1) \\ \mathbf{E}_f^{-1}(k)\mathbf{E}_{fJ}^T(k) \end{bmatrix} \tag{40}$$

The matrix $\mathbf{E}_{fJ}(k)$ is defined as,

$$\mathbf{E}_{fJ}(k) = [x_1(k)\mathbf{I}_{J\times J} \quad \ldots \quad x_I(k)\mathbf{I}_{J\times J}] - \mathbf{F}_J^T(k-1)\mathbf{D}_{fq2,v}(k) \tag{41}$$

where $x_i(k)$ is the current input of the ith channel, and $\mathbf{F}_J^T(0) = \mathbf{0}_{RL\times J}$

Table 2: The algorithm for computing $\mathbf{y}_J(k)$ in MFX ANC using MIMO-FQRD-RLS algorithm.

Initialize:
$\mathbf{F}_J(k-1) = \mathbf{0}_{(RL)\times J}$
Available from MIMO-FQRD-RLS algorithm:
$\mathbf{D}_{fq2}(k)$, $\mathbf{d}_{q2}(k)$, $\mathbf{E}_f(k)$
for each k
{
Obtaining $\mathbf{E}_{fJ}(k)$: using (41)
Obtaining $\mathbf{F}_J(k)$: using (40)
Obtaining $\mathbf{y}_J(k)$: using (36)
}

The lemma follows by taking the inverse transpose of (30) and post-multiplying it with $\mathbf{X}_J(k)$, which is a straightforward extension of the single-channel case detailed in [5].

The algorithm for reproducing the output $\mathbf{y}_J(k)$ in the upper branch of Fig. 2 is given in Table 2. The computational complexity of this method is mentioned in Table 3.

4. SIMULATIONS

This section provides the simulation setup and the results for the MIMO-FQRD-RLS algorithm applied to a multichannel active noise control system using MFX structure. To illustrate the MFX MIMO-FQRD-RLS algorithm we consider an ANC setup with primary paths and secondary paths are given as, $P_1(z) = z^{-3} - 1.8z^{-4} + 0.2z^{-5} + 0.1z^{-6}$, $P_2(z) = z^{-3} - 1.2z^{-4} + 0.5z^{-5} + 0.05z^{-6}$, $S_{11}(z) = 2z^{-2} + 0.5z^{-3} + 0.1z^{-4}$, $S_{12}(z) = 2z^{-2} + 0.3z^{-3} + 0.1z^{-4}$, $S_{21}(z) = z^{-2} - 1.7z^{-3} + 0.2z^{-4}$, $S_{22}(z) = z^{-2} + 0.2z^{-3} + 0.2z^{-4}$, where $P_r(z)$ denotes the primary path and $S_{jm}(z)$ denotes the secondary path, with $R = 2$, and $M = 2$. The order of the adaptive filter is $L = 49$, and the input signal $x(k)$ was a colored noise sequence, colored by filtering a zero-mean white Gaussian noise sequence $n_x(k)$ through a first order IIR filter $x(k) = -0.95x(k-1) + n(k)$. The desired signal $d(k)$ was further disturbed by noise whose variance is set such that SNR is 60dB. The results were obtained by averaging and smoothing 25 realizations of the experiment. For comparison purposes, a QRD-RLS algorithm is also implemented. Fig. 3 shows the MSE curves for both the algorithms. The results show that both algorithms converge to the same solution.

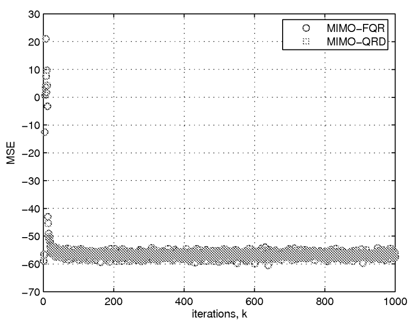

Figure 3: The MSE for MIMO-FQRD-RLS and MIMO-QRD-RLS algorithms in MFX structured ANC.

5. CONCLUSIONS

This paper derives a MIMO version of FQRD-RLS algorithm for application in MFX-ANC systems. It further shows how to avoid the explicit use of the weight vector in the MFX-ANC structure by reproducing the filtered-x signal from the internal variables of the proposed algorithm. The presented technique enables application of the MIMO-FQRD-RLS algorithms in MFX-ANC structure. The proposed method was verfied by simulating the multichannel MFX-ANC system. The results were compared with those using a design based on the QRD-RLS algorithm. It was verified that identical results are obtained using the proposed design method at a much lower computational cost.

Table 3: Operations required for MIMO-FQRD-RLS algorithm in MFX structure and Weight extraction method.

ALG.	MULT	DIV	SQRT
MIMO-FQR	$2R^3L + 4R^2L(M+1)$ $+2R^2M$ $+2RLM + 5RM + RL$	R^2L $+RLM$ $+RM$	$(R)^2L$ $+RLM$ $+RM$
MIMO-QRD	$4M(RL)^2$ $+4MRL + 2RL + 1$	$2MRL$	MRL
MFX-FQR	$2R^3L + 4R^2L(M+1)$ $+4R^2LJ + 2R^2M$ $+2RLM + 5RM + RL + 3RLJ$	R^2L $+RLM$ $+RM$	$(R)^2L$ $+RLM$ $+RM$
MFX-QRD	$4M(RL)^2$ $+4MRL + RLJ + RL + 1$	$2MRL$	MRL

Acknowledgment

This work was partially funded by the Academy of Finland, Smart and Novel Radios (SMARAD) Center of Excellence, PSATRI, CNPq, CAPES, and FAPERJ.

REFERENCES

[1] P. S. R Diniz, *Adaptive filtering: Algorithms and Practical Implementations*, Third Edition, Springer, USA, 2008.

[2] M. Shoaib, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Solution to the weight extraction problem in FQRD-RLS algorithms," *Proc. ICASSP'2006*, Toulouse, France, May 2006.

[3] M. Shoaib, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Multichannel fast QR-decomposition RLS algorithms with explicit weight extraction," *Proc. EUSIPCO'2006*, Florence, Italy, 2006.

[4] M. Shoaib, S. Werner, J. A. Apolinário Jr., and T. I. Laakso, "Equivalent output-filtering using fast QRD-RLS algorithm for burst-type training applications," *Proc. ISCAS'2006*, Kos, Greece, May 2006.

[5] José Antonio Apolinário Jr. (Editor), *QRD-RLS Adaptive Filtering*, Springer, 2009.

[6] M. Bouchard, "Numerically stable fast convergence least-squares algorithms for multichannel active sound cancellation systems and sound deconvolution systems," *Signal Processing*, vol. 82, no. 5, pp. 721-736, May 2002.

[7] A. L. L Ramos, J. A. Apolinário Jr, and S. Werner, "Multichannel fast QRD-RLS adaptive filtering: Block-channel and sequential algorithm based on updating backward prediction errors," *Signal Processing*, vol. 87, pp. 1781–1798, July 2007.